Research Note 85-87

COMBINING DECISION ANALYSIS AND ARTIFICIAL INTELLIGENCE TECHNIQUES: AN INTELLIGENT AID FOR ESTIMATING ENEMY COURSES OF ACTION

Paul E. Lehner, James R. McIntyre, Leonard Adelman, Leonard Adelman, Kermit Gates, Peter Luster, Matthew Probus and Michael L. McDonnell

PAR TECHNOLOGY CORPORATION

AD-A159 846

for

Contracting Officer's Representative
Beverly G. Knapp

Battlefield Information System Technical Area

SYSTEMS RESEARCH LABORATORY
Franklin L. Moses, Acting Director

U. S. Army
Research Institute for the Behavioral and Social Sciences

August 1985

Approved for public release; distribution unlimited.

85 10 7 088

U. S. ARMY RESEARCH INSTITUTE

FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency under the Jurisdiction of the

Deputy Chief of Staff for Personnel

EDGAR M. JOHNSON
Technical Director

WM. DARRYL HENDERSON
COL, IN
Commanding

A-1

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: Research Note 85-87	2. GOVT ACCESSION NO.: AD-A159 846
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): COMBINING DECISION ANALYSIS AND ARTIFICIAL INTELLIGENCE TECHNIQUES: AN INTELLIGENT AID FOR ESTIMATING ENEMY COURSES OF ACTION	5. TYPE OF REPORT & PERIOD COVERED: Final Report, Jan. 1984-March 1985
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Paul E. Lehner, James R. McIntyre, Leonard Adelman, Kermit Gates, Peter Luster, Matthew Probus and Michael L. Donnell	8. CONTRACT OR GRANT NUMBER(s): MDA903-83-C-0311
9. PERFORMING ORGANIZATION NAME AND ADDRESS: PAR TECHNOLOGY CORPORATION/DSS Section, Suite 840, 7926 Jones Branch Drive, McLean, Virginia 22102	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 2Q162717A790
11. CONTROLLING OFFICE NAME AND ADDRESS: ARMY RESEARCH INSTITUTE - PERI-SFA, 5001 Eisenhower Avenue, Alexandria, VA 22333	12. REPORT DATE: August 1985
	13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): --	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

--

18. SUPPLEMENTARY NOTES

Beverly G. Knapp, contracting officer's representative
Mary Jo Hall, reviewer

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

EXPERT SYSTEMS
HUMAN FACTORS
MAN/MACHINE INTERFACE

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

AI/ENCOA is a prototype decision aid designed to assist Army tactical intelligence analysts in evaluating alternative Enemy Courses of Action. AI/ENCOA combines the use of additive Multi-attribute Utility (MAU) models for course of action evaluation with rule-based procedures for assigning parameter values (scores and weights) to the MAU model.

(continued on next page)

ITEM 20 (continued)

AI/ENCOA is composed of two parts: a generic software package that implements a combined AI/MAU architecture, and two COA 'rule bases' for evaluating different types of possible enemy COAs.

The present final report summarizes the technical progress made in developing AI/ENCOA.

FOREWORD

AI/ENCOA (Artificial Intelligence/ENemy Courses Of Action) is a prototype decision aid designed to assist Army tactical intelligence analysts in evaluating alternative Enemy Courses of Action. AI/ENCOA combines the use of additive MAU (Multi-attribute Utility) models for course of action evaluation with rule-based procedures for assigning parameter values (scores and weights) to the MAU model.

AI/ENCOA is composed of two parts: a generic software package that implements a combined AI/MAU architecture, and two COA 'rule bases' for evaluating different types of possible enemy COAs.

The present final report summarizes the technical progress made in developing AI/ENCOA.

COMBINING DECISION ANALYSIS AND
ARTIFICIAL INTELLIGENCE TECHNIQUES:
AN INTELLIGENT AID FOR ESTIMATING ENEMY COURSES OF ACTION

EXECUTIVE SUMMARY

Requirement:

To develop a prototype computerized aid for conducting U.S. Army tactical intelligence analyses that utilizes state-of-the-art computerized support, such as artificial intelligence techniques, and is implemented in the PASCAL language for use on a government-owned IBM Personal Computer micro-processing system.

Procedure:

AI/ENCOA is a prototype decision aid designed to assist Army tactical intelligence analysts in evaluating alternative enemy courses of action. It was produced by combining the use of additive multi-attribute utility (MAU) models for course-of-action evaluation with rule-based procedures for assigning parameter values (scores and weights) to the MAU model.

Findings:

AI/ENCOA is composed of two parts: a generic software package that implements a combined Artificial Intelligence (AI)/MAU architecture, and two Course of Action (COA) 'rule bases' for evaluating different types of possible enemy COAs. The rule bases may be altered without reprogramming the software.

Utilization of Findings:

1. AI/ENCOA can generate solutions to certain "textbook" problems and, therefore, may be appropriate as instructional support at the U.S. Army Intelligence Center and School. 2. By altering the rule bases, one can enter a variety of different problems, and generate the solutions to these problems, thereby using AI/ENCOA to potentially provide cognitive support to users in a number of different tactical intelligence analysis areas.

Table of Contents

List of Figures

1.0 INTRODUCTION

Techniques from the disciplines of Artificial Intelligence (AI) and Decision Analysis (DA) have both been extensively used in the development of computerized decision aids. In AI, rule-based program architectures have been instrumental in the implementation of expert systems that serve as knowledgeable consultants in a variety of problem domains. In DA, normative decision aids have been developed that use prescriptive problem representations to help guide users through the decision making process.

Unfortunately, from the perspective of many types of practical decision aiding applications, both normative decision aids and expert system technology have significant limitations. In particular, in expert system development there is a lack of established techniques for problem structuring and knowledge engineering. This usually leads to time-consuming rule base development efforts with limited success in domains where the knowledge required to solve problems is not already well documented (Davis, 1982). Normative decision aids, on the other hand, are usually built around a prescriptive, but rigid problem structure called a decision analysis model that may not be compatible with the "evolutionary" approach to system development that is characteristic of AI.

This report outlines a practical approach to decision aid development that systematically utilizes both the problem structuring techniques of DA and the incrementally modifiable software architectures found in AI. The approach advocates the use by knowledge engineers of DA modeling techniques for the initial

structuring of expert knowledge, while at the same time it advocates the use of AI software architectures that separate domain knowledge from general problem solving procedures. A specific instantiation of this approach is presented. This system is a decision aid for evaluating ENemy Courses Of Action (ENCOA) within a rule-based program architecture that is referred to here as AI/ENCOA. The decision analytic model in AI/ENCOA is based, in part, on a previous ENCOA aid that utilized a more conventional program architecture.

The previous ENCOA aid, like all normative decision aids developed up to that time, required of users that they fully understand how to implement the aid's decision-theoretic approach. Perhaps more importantly, the description within the aid of the operational environment often had to be modified in order to permit users to implement the aid's decision theoretic approach. As a result, it frequently happened that potential users, even though convinced of the aid's utility, would not use the aid.

For example, tactical intelligence analysts using ENCOA had continually to re-evaluate (1) the appropriateness of the attributes in the MAU model, (2) the scores for each alternative on all appropriate attributes, and (3) the weights indicating the relative importance of the differences between the best and worst alternatives on the attributes. Consequently, ENCOA required that the decision making process, at a minimum, give analysts the time necessary to implement these steps. An informal evaluation of ENCOA by Army personnel at Fort Bragg suggested that tactical intelligence analysts might resist learning MAU analysis and/or modifying the operational environment description sufficiently to use ENCOA, even though the aid

appeared to them to have considerable practical value.

In contrast to the previous ENCOA aid, AI/ENCOA interacts with the user through a built-in Attribute Manager. The Attribute Manager asks the user a series of questions about the military situation. The user can answer these questions with very simple responses, such as Yes (y), No (n), or Don't Know (Carriage return). Each question corresponds to an attribute in a predefined attribute list. User answers to the questions set the 'truth value' for each attribute in the attribute list. Presence of the Attribute Manager thus turns AI/ENCOA into a "consultation system". All information about the user's specific problem and the military situation is obtained by querying the user directly through the Attribute Manager.

The way the Attribute Manager interacts with the user may itself be modified, without reprogramming, by changing the AI/ENCOA rule base. This observation suggests two additional ways in which AI/ENCOA differs from its ENCOA predecessor. First, the AI/ENCOA software is entirely generic, allowing users to develop or tailor models for any type of model selection domain. In contrast, ENCOA was specific to the COA [Courses Of Action (see Section 3.0)] problem. And secondly, the original ENCOA addressed only the problem of evaluating alternative avenues of approach (AOAs). AI/ENCOA, in contrast, contains two models. The first model evaluates, at the division level, whether the enemy commander might engage in a Primary Attack, Secondary Attack, Defense, or Withdrawal. The second model discriminates between primary and secondary AOAs, given that some form of attack will occur.

The remainder of this report is organized as follows. Section

2.Ø below summarizes, in general terms, our approach to combining decision analysis and artificial intelligence techniques in a decision aid. Section 3.Ø provides a technical overview of AI/ENCOA along with an annotated copy of an excerpt from an interactive session using it. Section 4.Ø provides a summary of where AI/ENCOA stands today, how it compares to the original ENCOA, and discusses options for further development.

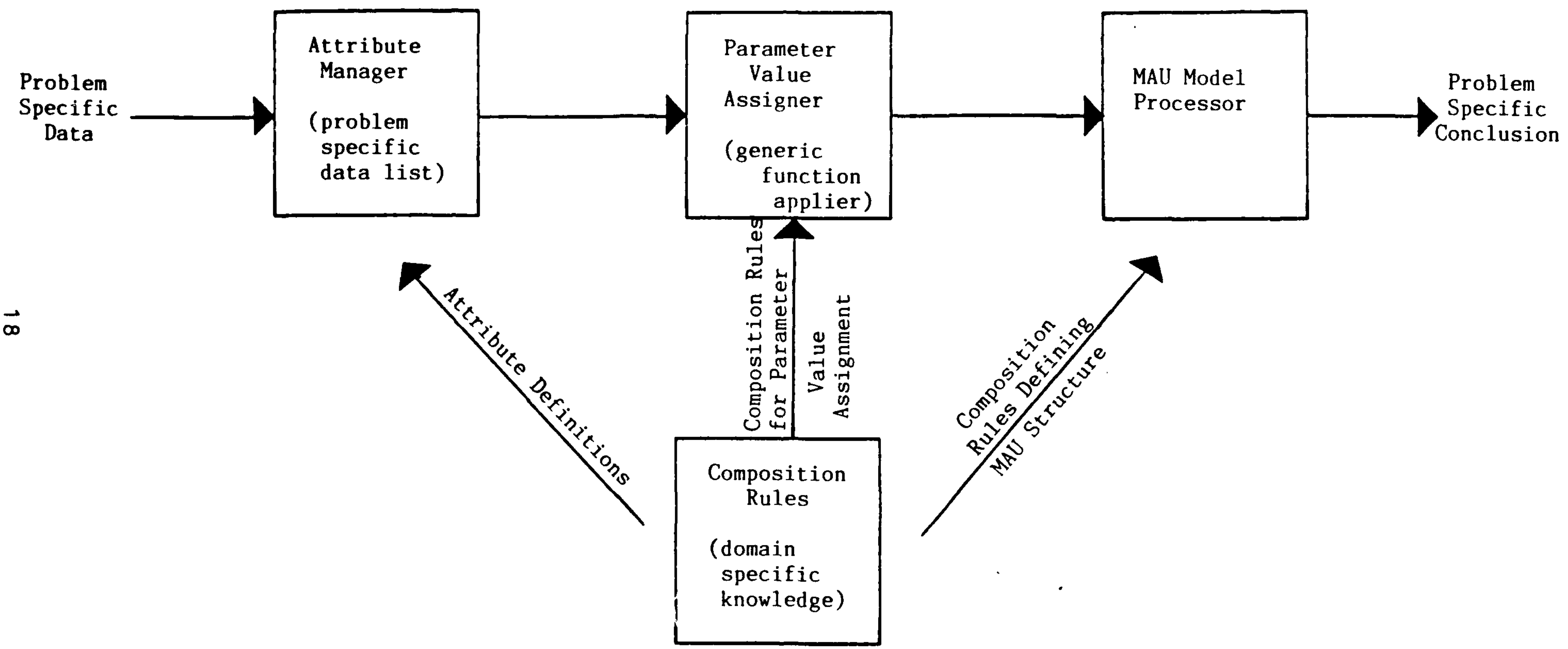


FIGURE 3-1

AI/ENCOA ARCHITECTURE

3.0 AI/ENCOA: A COMBINED ARTIFICIAL-INTELLIGENCE/ DECISION-ANALYSIS DECISION AID

AI/ENCOA is a prototype decision aid designed to assist Army tactical intelligence analysts in evaluating alternative Enemy Courses of Action (COAs). AI/ENCOA combines the use of additive MAU models for course of action evaluation with rule-based procedures for assigning parameter values (scores and weights) to the MAU model.

Functionally, AI/ENCOA can be composed of two parts: a generic software package that implements a combined AI/MAU architecture, and two COA 'rule bases' for evaluating different types of possible enemy COAs. Section 3.1 below provides a technical overview of the generic software. Section 3.2 overviews the two COA models. Section 3.3 provides an excerpt from a session with AI/ENCOA.

3.1 A Combined Artificial-Intelligence/Multi-Attribute-Utility Architecture

Conceptually, the AI/MAU software has three interacting components: (1) an MAU model and analysis capability; (2) a user interface system, called the Attribute Manager, that permits users to characterize the decision situation facing them, and (3) a set of composition or production rules that translate the description of the situation into appropriate scores and weights in the MAU model, thereby tailoring the MAU model to the specifics of the present problem. (See Figure 3-1.)

The role of the Attribute Manager is to query the user as to the nature of the decision situation. The Attribute Manager asks the user a series of questions about the specific problem the user is

composition rules define the composition trees on the left side. Although computationally simple, this approach has some strong practical advantages. The most important of these advantages is that it supports a model/knowledge base development and enhancement process that (1) can start with a strict DA model for the first cut composition trees, but (2) can allow for incremental modification and enhancement of the initial model, because it allows modifications to individual nodes.

As noted above, DA provides a number of procedures for model development that usually result in first cut composition trees that approximate a normative structure. Consequently, these procedures provide an effective approach to generating a first cut at composition trees that are not likely to require significant reorganization of the problem structure during later stages of the aid development process. However, fine tuning of the model as a result of feedback can still be done in the same manner as with most expert systems -- viz, by testing the system and then modifying individual rules to improve the test results, and so on, iteratively, until the system tests satisfactorily.

The next section describes a decision aid that represents a specific instantiation of this combined AI/DA approach to aid development.

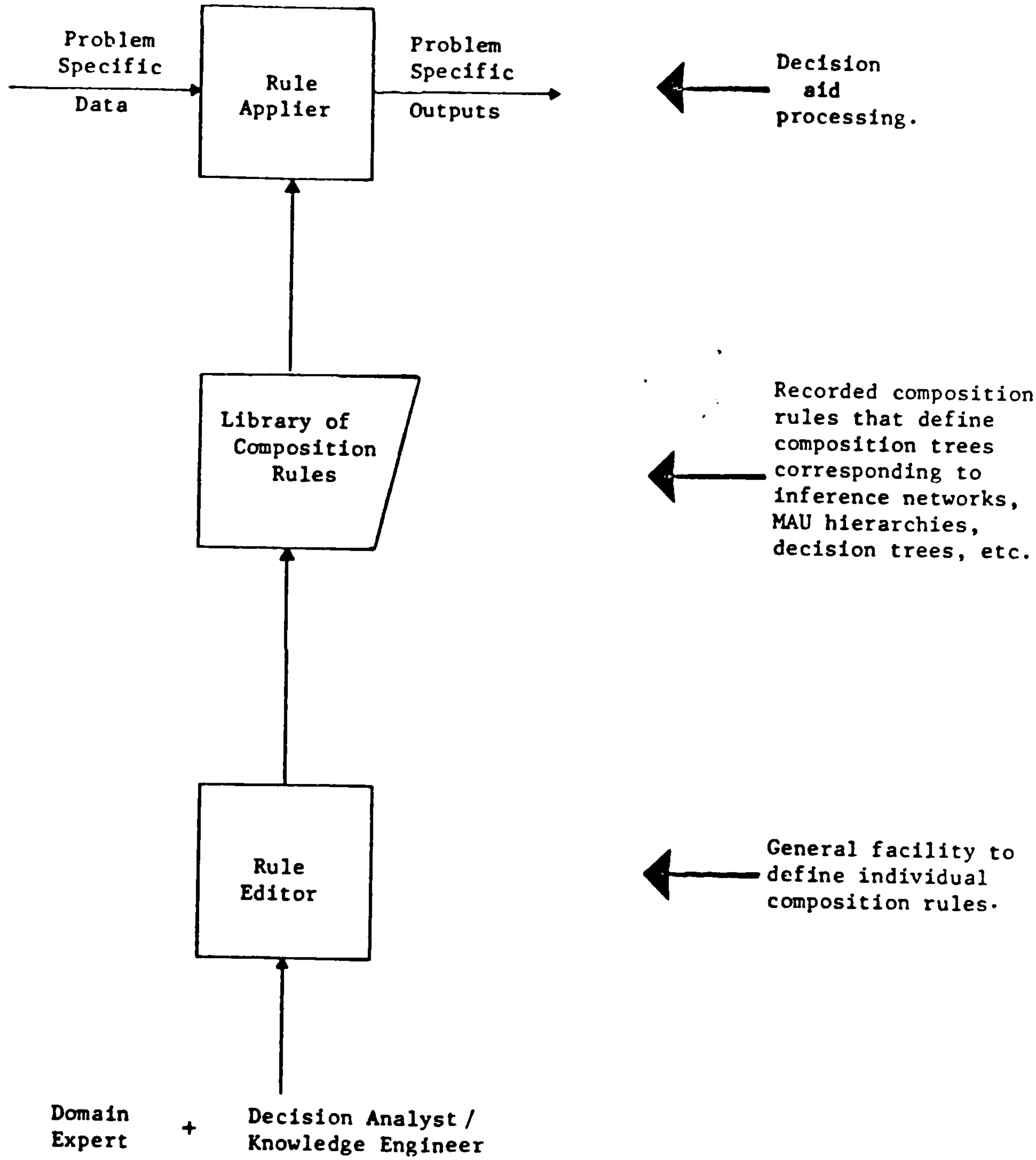
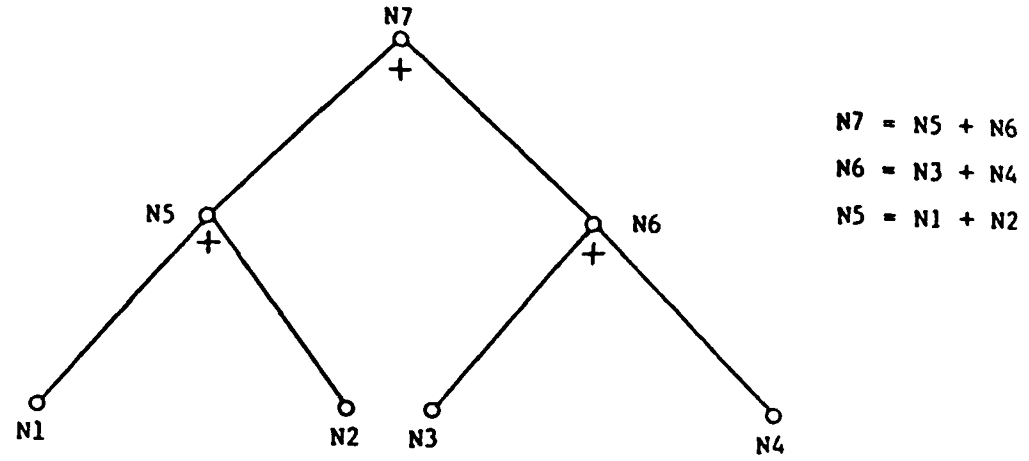
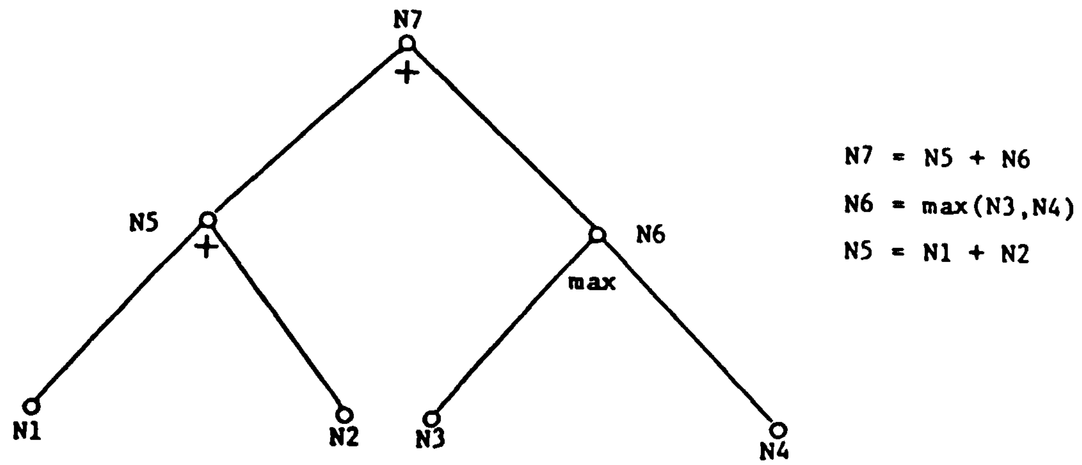


FIGURE 2-5

A COMBINED AI/DA ARCHITECTURE



4A - Initial Structure



4B - Revised Structure

FIGURE 2-4

TWO SAMPLE COMPOSITION TREES WITH
CORRESPONDING COMPOSITION RULES

instance, going from the structure in Figure 2-4A to the one in Figure 2-4B would involve reprogramming within many DA-based decision aids. Third, DA aids, as with expert systems, normally require users to subjectively assign values to the bottom-level factors. However, with DA aids, this may become problematic since a common by-product of problem structuring via DA models is defining a set of primitive factors different from the set of problem elements normally perceived by the user. The user is therefore required to translate his or her perception of the problem into the input requirements of the DA-based decison aid.

Combining DA and AI Techniques in Decision Aid Development

There is a natural synergy between the prescriptive problem structuring techniques in decision analysis and the rule-based program architectures used in AI expert systems. In particular, from the perspective of building decision aids, DA modeling procedures are suitable for problem structuring while AI rule-based program architectures are suitable for (1) making the problem structure incrementally modifiable, and (2) developing a user interface that uses only terms and references familiar to users. The basic approach to building aids that take advantage of this synergy is to build software modules that do not assume a priori limitations on the form of the decision model, but rather allow model definition to be an incrementally modifiable portion of the system. This is done by encoding a model as a set of separate composition rules that can be individually added, deleted, or modified by a general rule editor (See Figure 2-5). In Figure 2-4, for instance, the right hand side

affected by changes to other parameter values in the model, then the use of the MAU model is normatively correct for a problem domain in which this 'value independence' axiom is satisfied (see Keeney and Raiffa, 1976, Chapter 3, for formal definitions and discussion). Consequently, representing expert knowledge becomes a process of developing decision models that reflect problem decompositions that satisfy the axioms of a normative decision model.

The key to the practicality of this technique as an approach to modeling expert knowledge is that most of these axioms can be tested within the context of interactive working sessions between the domain expert and the decision analyst. This makes it relatively easy to iterate through several cycles of problem restructuring prior to encoding the expert model into computer usable form.

Using decision models for decision aid development, however, has historically presented some difficulties. The first problem is the fact that there is a significant difference between building a normative decision model for a single problem, and the repetitive use of a template decision model across multiple problems within a domain. In the first case, in order to guarantee that the axioms are satisfied, the model can be carefully tailored, often in an ad hoc manner to the specifics of the problem at hand. In the latter case, the model remains static across applications. A static template model can at best be only a first approximation to a normative, problem-specific structure. A second problem is the fact that decision aids using decision analytic models are normally implemented using conventional hierarchical programming structures which place severe limits on the modifiability of the problem structure. For

how such a methodology may be borrowed from the techniques of DA.

Aids Using Normative Decision Models from Decision Analysis

Over the last twenty-five years, hundreds of scientific studies of human judgment and decision making have shown that unaided human judgment has limitations (Hammond, McClelland, Mumpower, 1980). As a result of these findings, as well as advances in the development of normative decision theory (Keeney and Raffai, 1976) and computer technology, computer-based decision aids have been developed that use normative decision models to organize and support decision making processes. These include a number of aids based on multi-attribute utility (MAU) models (Adelman, Donnell, Phelps, 1981; Hammond, Cook, Adelman, 1977), as well as more traditional decision-analytic models that combine probability and utility assessments, (Steeb & Johnson, 1981).

In normative decision aids, these normative models operate, in effect, as prescriptive problem structures that serve to provide an approach to organizing and using expert knowledge. Indeed, many stand alone aids of this type are designed primarily to step a user through an axiomatically correct process for using his or her own knowledge to solve a problem.

Normative decision models are based on axioms from decision theory and measurement theory, which guarantee that if the axioms are satisfied in a problem domain, then the problem decomposition and the form of the corresponding composition equations are necessarily correct. For example, for additive MAU models it has been shown that if the value contributed by any element in the MAU model is not

TYPE OF RULE	DEGREE OF BELIEF CALCULATION
AND	Deg(H1) = min(Deg(E1), Deg (E2))
OR	Deg(H6) = max(Deg(H1), Deg(H2))
NOT	Deg(H7) = - Deg(H3)
Modified Bayesian	Functions derived from Bayes Rule: For example in AL/x (Reiter, 1981) Degree (H3) = W + Deg(E5), where W is calculated by a linear interpolation on Deg(E5) between the positive and negative weights, pw and nw, linking E5 to H3.

FIGURE 2-3

SOME COMMON DEGREE OF BELIEF PROPAGATION FUNCTIONS

associated prior degree of belief and a rule for combining subnode belief values into an updated degree of belief for the node. Example combination rules, using the structure in Figure 2-2, are listed in Figure 2-3.

Knowledge engineering, the process of abstracting and encoding human expertise, can be viewed as the process of generating a set of inference networks appropriate to a problem domain. Unfortunately, regarding this process, at present there appears to be a lack of established approaches to problem representation and decomposition, (i.e., constructing inference networks). As a result, the development of a knowledge base is often a very time-consuming part of building an expert system (Davis, 1982). In particular, what can occur is that the initial versions of a rule base will reflect a poor problem representation, which results in a need for a considerable modification and restructuring of the networks. It is usually only after several iterations on the organization of the knowledge base that an expert system will begin to "look smart." [Indeed, specified knowledge engineering procedures that do presently exist seem to establish the need for iterative restructuring (e.g., Buchanan et al., "Constructing an Expert System).]

These shortcomings of the traditional knowledge engineering approaches suggest the need for some new methodology. The new methodology will, with high probability, permit the quick and efficient construction of a "first cut" knowledge base that approximates the finished product -- i.e., that can evolve into the finished knowledge base through a process more akin to "fine tuning" than to successive "radical reconstructions". The sequel will suggest

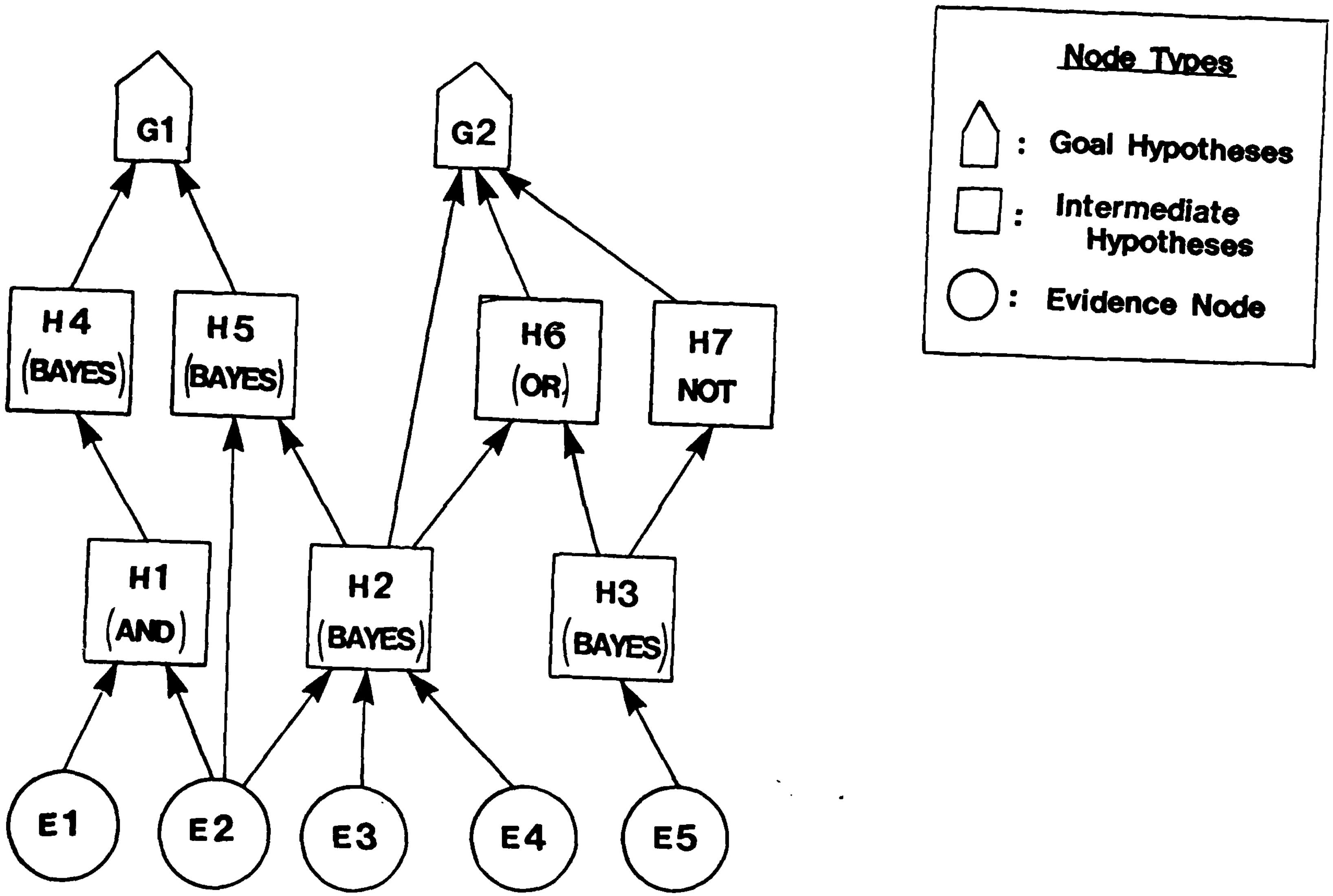


FIGURE 2-2

SAMPLE FORM OF AN INFERENCE NETWORK

order to generate problem specific conclusions.

The major advantage of a rule-based program architecture, as compared to more conventional hierarchically organized programs, is that it permits an evolutionary approach to system development. That is, once general decisions have been made regarding the basic control procedures and the organization of the rule-base, the knowledge base can be incrementally improved by adding, modifying, or deleting individual production rules. In more conventional structures, changing the problem solving procedures often requires a substantial, and time-consuming, modification of existing programs, data structures, and sub-routine organization. A second advantage of encoding knowledge in the form of production rules is that it makes it relatively easy to develop a user interface containing only terms and references familiar to the user. In particular, since the various rule preconditions correspond to problem attributes that human experts have identified in the knowledge engineering process, it is easy to write queries that ask users about the status of these problem attributes.

Most expert systems deal with various classes of inference problems, where the expert system must draw conclusions from various evidence/data inputs. In these types of inference problems, the set of rules in a rule-base can be graphically represented in the form of a set of inference networks. As illustrated in Figure 2-2, an inference network contains top-level hypotheses, called goal hypotheses, which are decomposed into various levels of subhypotheses that are further broken down into specific items of evidence that can support those hypotheses. With each node, there is usually an

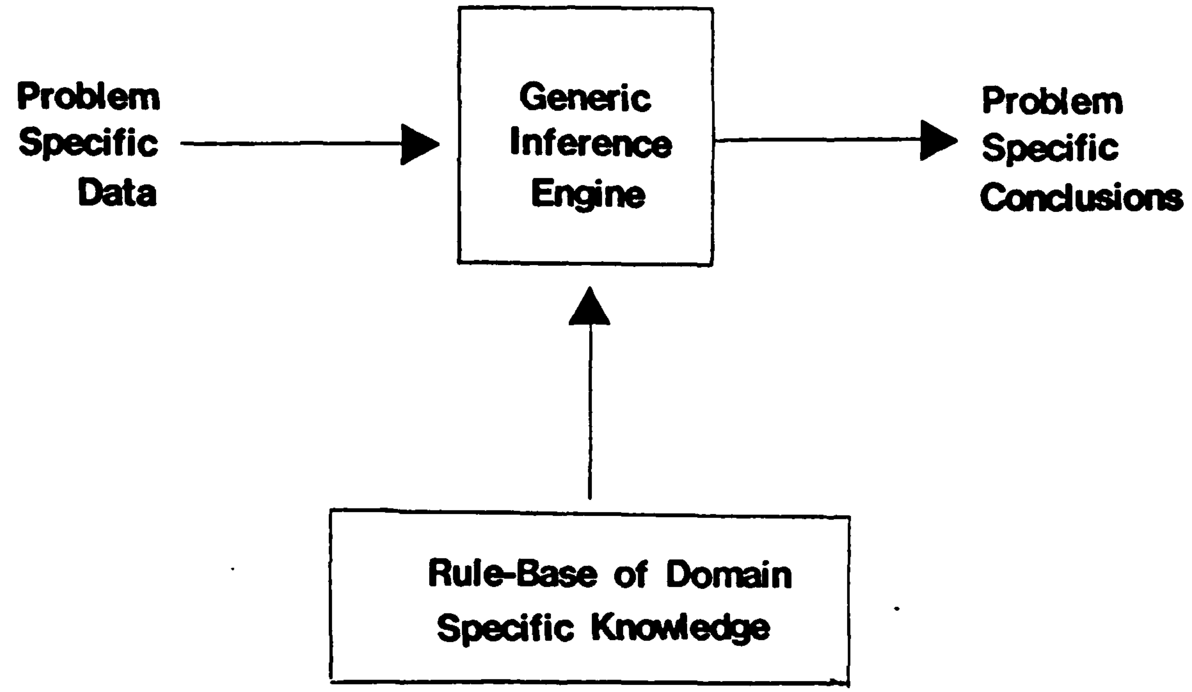


FIGURE 2-1

SIMPLIFIED OVERVIEW OF EXPERT SYSTEM ARCHITECTURE

2.0 ON THE SYNERGY BETWEEN DECISION ANALYSIS AND ARTIFICIAL INTELLIGENCE

The combined AI/DA approach in this paper uses the structure of a multi-attribute utility (MAU) aid and a rule-based procedure to reduce the typically tedious inputs to aid development. The two following subsections discuss the key aspects of AI and DA, respectively, that form the basis of the approach; readers interested in further details should consult the cited references. The third subsection discusses the combined AI/DA approach to aid development.

Rule-Based Systems In AI

Artificial Intelligence (AI) is a discipline dedicated to the development of computer systems which exhibit intelligent behavior. One important area within AI is the development of expert systems that serve as knowledgeable consultants in a variety of problem domains (Duda, Hart, Gashnig, 1977; Shortliffe, 1978; Buchanan, 1978).

These systems are composed of essentially two components, "a knowledge base" and an "inference engine". In the knowledge base, domain specific knowledge is expressed as a set of condition-action pairs referred to as production rules that specify the action to be carried out if the prerequisite conditions are true. (Frequently the 'action' is to modify the degree of belief in a hypothesis.) The role of the inference engine is to control the order of rule activation and to update the belief value of hypotheses being considered based upon acquired evidence. In effect, as shown in Figure 2-1, the inference engine applies domain specific knowledge to problem-specific data in

addressing. Each question corresponds to an attribute in a predefined attribute list. User answers to the questions set the value of each attribute in the general attribute list. Users also have the option to select and answer only those few questions addressing specific, minimal changes in repetitive decision situations, thereby permitting them to quickly modify the status of the attribute list.

The role of the parameter assignment rules is to translate the information about the decision situation, encoded in the attribute list, into scores and weights in the MAU model. This rule-base will be decomposed into independent rule sets that correspond to the nodes in the MAU hierarchy. For each node in the hierarchy there is a set of composition rules that determine the value of the parameters associated with that node. The preconditions in each rule correspond to one or more attributes in the attribute list. The action resulting from each rule is the assignment or functional adjustment of the parameter value of the associated node in the hierarchy.

Figure 3-2 shows a simple example of an attribute, four parameter assignment functions, and a terminal MAU factor drawn from the ENCOA rule base described in the next section. The attribute definition defines a multiple choice question that will be asked the user. Based on the user's response, the variable f_of_f will be assigned the value 1, 2, or 3. The values of the variables f_of_f1 through f_of_f4 are a function of f_of_f such that for the terminal factor (FIELDS_OF_FIRE) in the MAU model scores for the four options (primary_attack, secondary_attack, defend, and withdraw) will be equal to the values of f_of_f1 through f_of_f4 respectively. Figure 3-3 shows this functional relationship. Note also that the weight of the MAU factor

ATTRIBUTE DEFINITION

```
multiple_choice question

   f_of_f
   -> 'Fields of Fire'
   1: 'Greater than 3000 Meters'
   2: 'Between 1500 and 3000
         Meters'
   3: 'Less than 1500 Meters';
```

PARAMETERS ASSIGNMENT RULES

```
function
   f_of_f1=
   if f_of_f=1 then 10 else
   if f_of_f=2 then 80 else
   if f_of_f=3 then 100 else
   0;

function
   f_of_f2=
   if f_of_f=1 then 40 else
   if f_of_f=2 then 90 else
   if f_of_f=3 then 100 else
   0;

function
   f_of_f3=
   if f_of_f=1 then 100 else
   if f_of_f=2 then 90 else
   if f_of_f=3 then 30 else
   0;

function
   f_of_f4=
   if f_of_f=1 then 0 else
   if f_of_f=2 then 50 else
   if f_of_f=3 then 60 else
   0;
```

MAU FACTOR DEFINITIONS

```
factor

   FIELDS_OF_FIRE
   weight f_of_f
   parent terrain_factors
   primary_attack f_of_f1
   secondary_attack f_of_f2
   defend f_of_f3
   withdraw f_of_f4;
```

FIGURE 3-2

EXTRACTS FROM ENCOA RULE BASE

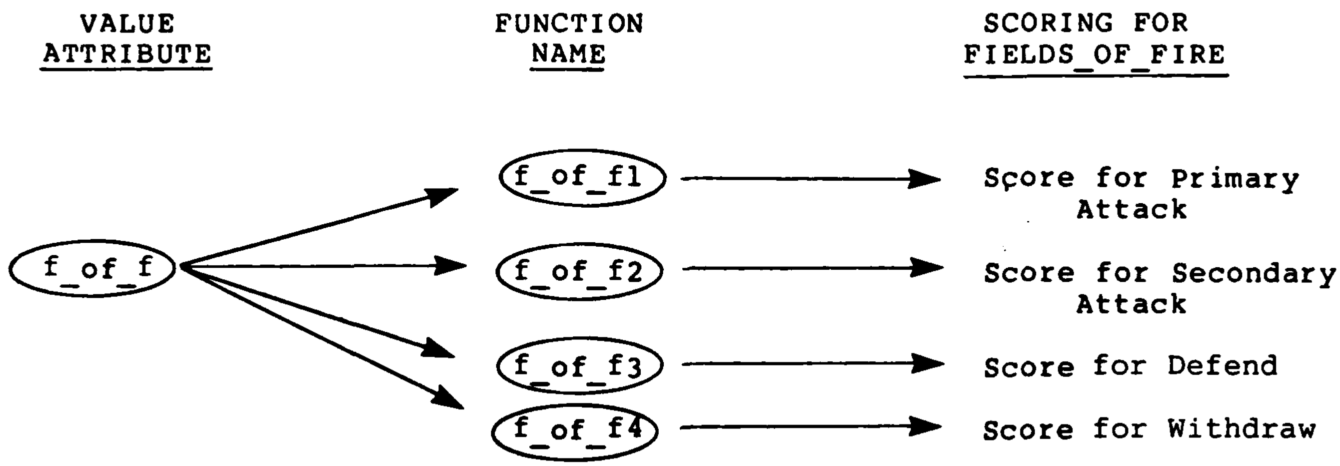


FIGURE 3-3

VARIABLE MAPPING IN EXTRACT FROM ENCOA RULE BASE

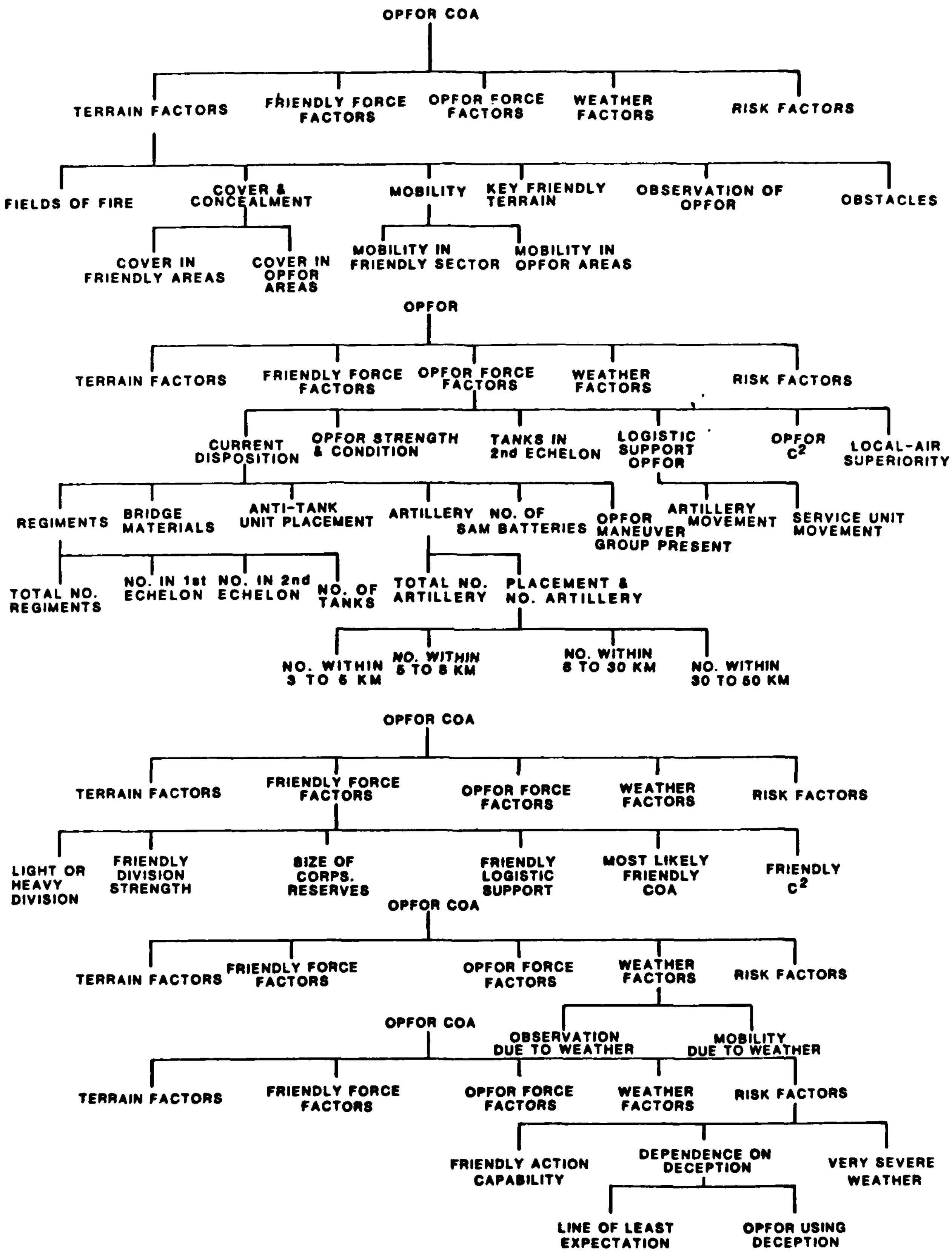
Fields_of_Fire is equal to the value of wf_of_f which could be, in turn, a function of the answer to other questions. A description of how to develop rule bases within the AI/MAU architecture is found in the Appendix.

This general approach to building an MAU-based expert system has two distinct advantages. First, the use of the Attribute Manager and parameter assignment rules make it possible to interface with users in terms and references with which they are familiar. In this regard, the user interface is very similar to that found in expert systems that do not contain a normative decision model. Second, as with other rule-based systems, this aid can be incrementally improved by simply adding, deleting, or modifying individual rules. This makes it possible to continually improve the aid's knowledge base, encoded as a combined normative MAU model and rule-base, over time.

3.2 Enemy Courses of Action Rule Bases

AI/ENCOA presently has available two 'rule bases' and models for evaluating possible enemy COAs. The first model addresses the question of whether the opposing forces facing a friendly division commander are likely to engage in a primary attack, secondary attack, remain in a defensive posture, or withdraw. This model is designed to help the intelligence analyst evaluate the severity of the threat that a friendly division commander may be facing. Figure 3-4 shows the MAU hierarchy corresponding to this first model.

The second model is desired to help the analyst examine the support for a Primary or Secondary Attack, along each of the different enemy avenues of approach (AOA) into a given friendly division sector



MAU STRUCTURE FOR COMPARING
PRIMARY ATTACK, SECONDARY ATTACK,
DEFENSE & WITHDRAW OPTIONS
FIGURE 3-4

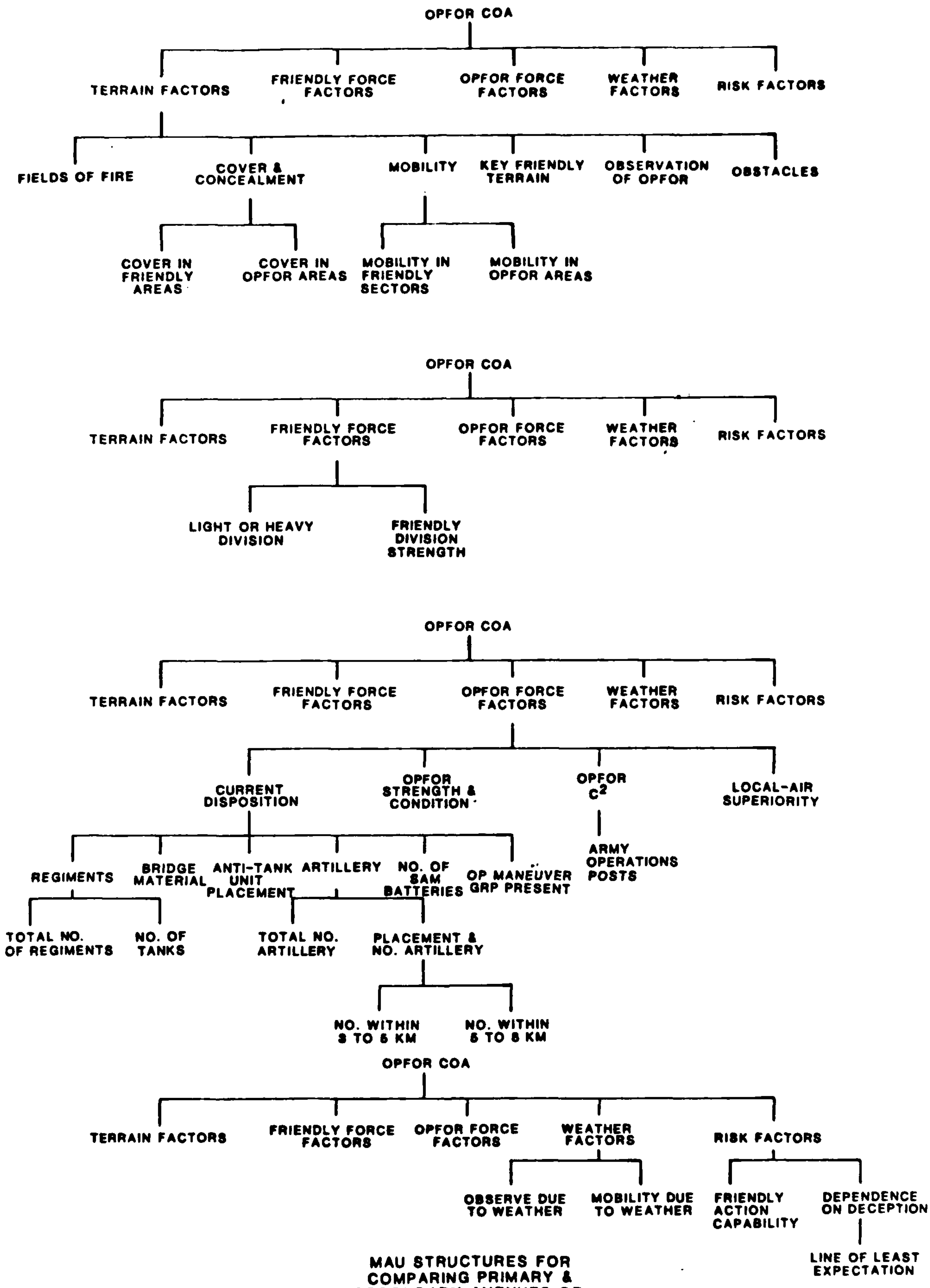
once it has been determined that it is likely that there will be a primary or secondary attack. That is, given that the analysis of the first model determines that a division commander is likely to be facing an attack of some type, then the AOA model helps determine the degree of support for a primary or secondary attack along each of the adversary lines of advance. The MAU structure for the AOA evaluation model is shown in Figure 3-5.

3.3 Excerpt from AI/ENCOA Session

The purpose of showing this excerpt is to give the reader a general flavor of how users interact with AI/ENCOA and to show how the components of AI/ENCOA discussed in the previous two sections fit together. Consequently, no attempt has been made to show all the capabilities; there is a separate user's manual which contains a complete AI/ENCOA training session (Luster et al., 1985).

Figures 3-6 through 3-11 are annotated hardcopies of several displays from an interactive session with AI/ENCOA. In paging through these figures, the reader will see a 'single thread' example of managing attributes (Figures 3-7 and 3-8), assigning scores to options (Figure 3-9), and a textual result display of the Enemy COAs that received the strongest support (Figures 3-10 and 3-11).

For each of these figures, the top half shows the AI/ENCOA display and the bottom half a brief annotation. Each AI/ENCOA display contains several windows. The upper window is the primary output display. The bottom half of the AI/ENCOA provides windows that (1) describe the characteristics and status of the MAU factor presently being examined (the "Description" window), (2) show the scores for



MAU STRUCTURES FOR COMPARING PRIMARY & SECONDARY AVENUES OF APPROACH IN A DIVISION SECTION

FIGURE 3-5

Network Structure Diagram

```
                          |-TERRAIN FACTORS---------|
                          |                         | |-FIELDS OF FIRE*
                          |                         | |
                          |-FRIENDLY FORCE FACTORS  | |-COVER AND CONCEALMENT
                          |                         | |
                          |                         | |-MOBILITY
OPFOR COA-----------------|-OPFOR FORCE FACTORS     |-|
                          |                           |-KEY FRIENDLY TERRAIN*
                          |                           |
                          |-WEATHER FACTORS           |-OBSERVATION OF OPFOR*
                          |                           |
                          |                           |-OBSTACLES*
                          |-RISK FACTORS

                    TERRAIN FACTORS - Subfactor of OPFOR COA
|-Description-----------------|ECOA Scores-------------------------------------|
| Wgt: 0.15   #Subfactors:  6 |  PRIMARY    SECONDARY    DEFEND    WITHDRAW     |
|%Sol:   0     #Ans Ques:   0 |     0           0           0          0        |
| Sec:sect 1  #Unans Ques:  8 |                                                 |
|-Solution Menu-----------------------------------------------------------------|
|  <Esc> Quit Menu       1. Answer Questions        4. Modify Sectors           |
|     <?> HELP           2. Display Results         5. Save/Restore Answers     |
|<PrtSc> Print Screen    3. Analyze Results                                     |
|-------------------------------------------------------------------------------|
```

FIGURE 3-6

In these excerpts, we will only consider the subproblem of evaluating the options of primary attack, secondary attack defense or withdrawal from the perspective of TERRAIN FACTORS. The above display shows that we have moved down to the TERRAIN FACTORS part of the model.

Answering of Questions

```
Characterize Fields of Fire as:
    1 : Greater than 3000 Meters
    2 : Between 1500 and 3000 Meters
    3 : Less than 1500 Meters
2
--------------------------------------------------------------------------------
Characterize Cover and Concealment into Friendly Sector as:
    1 : Many (3 or More) Totally Covered and Concealed Routes
    2 : Few (1 or 2) Covered and Concealed Routes
    3 : Partially Covered and Concealed Routes
    4 : No Covered and Concealed Routes
```

```
                    TERRAIN FACTORS - Subfactor of OPFOR COA
|-Description-----------------|ECOA Scores-------------------------------------|
| Wgt: 0.15   #Subfactors:  6 |  PRIMARY    SECONDARY    DEFEND    WITHDRAW    |
|%Sol:    0     #Ans Ques:  0 |     0           0           0          0       |
| Sec:sect 1  #Unans Ques:  8 |                                                |
|-Question Menu:-------------------------------------------------------------|
|  <Esc> Quit Menu     1. Answer All Questions   4. Display Answers          |
|     <?> HELP         2. Unanswered Questions   5. Erase Answers            |
|<PrtSc> Print Screen 3. Modify Answers                                      |
|----------------------------------------------------------------------------|
```

FIGURE 3-7

There are a total of eight questions relevant to evaluating TERRAIN FACTORS. This figure shows a display generated during the process of answering these questions. Note that the top question corresponding to the attribute shown in Figure 3-2.

Display of Answers

```
Question Regarding: Fields of Fire
    Current Status: Between 1500 and 3000 Meters
-----------------------------------------------------------------------------
Question Regarding: Cover and Concealment into Friendly Sector
    Current Status: Few (1 or 2) Covered and Concealed Routes
-----------------------------------------------------------------------------
Question Regarding: Cover and Concealment About OPFOR Assembly Areas
    Current Status: No Covered Assembly Areas
-----------------------------------------------------------------------------
<More>
```

```
                  TERRAIN FACTORS - Subfactor of OPFOR COA
|-Description-----------------|ECOA Scores-----------------------------------|
| Wgt: 0.15  #Subfactors:  6 |  PRIMARY   SECONDARY   DEFEND    WITHDRAW    |
|%Sol:  100    #Ans Ques:  8 |    40         64         66         30       |
| Sec:sect 1  #Unans Ques:  0 |                                              |
|-Question Menu:----------------------------------------------------------------|
|  <Esc> Quit Menu    1. Answer All Questions   4. Display Answers          |
|     <?> HELP         2. Unanswered Questions   5. Erase Answers            |
|<PrtSc> Print Screen 3. Modify Answers                                     |
|----------------------------------------------------------------------------|
```

FIGURE 3-8

The output window above shows the status of some of the attributes. Note that the Description window show that all eight questions have been answered. Also, the ECOA score window shows the resulting score for each option on TERRAIN FACTORS.

Summary of Scores: Numerical Display

Factor Name	Degree Solved	Rel. Weight	PRIMARY	SECONDA	DEFEND	WITHDRA
			ECOA Scores			
Current Factor:						
FIELDS OF FIRE	100%	0.17	80	90	90	50

```
FIELDS OF FIRE - Bottom-Level Factor, Subfactor of TERRAIN FACTORS
|-Description----------------|ECOA Scores------------------------------------|
| Wgt: 0.17    #Subfactors:  0 |  PRIMARY    SECONDARY    DEFEND    WITHDRAW   |
|%Sol:  100    #Ans Ques:    1 |   80         90          90         50       |
| Sec:sect 1   #Unans Ques:  0 |                                              |
|-Score Summmarization Menu--------------------------------------------------|
|  <Esc> Quit Menu     1. Textual Summary           4. Cross Sector Summary |
|    <?> HELP          2. Tabular Summary                                   |
|<PrtSc> Print Screen 3. Numerical Summary                                  |
|---------------------------------------------------------------------------|
```

FIGURE 3-9

The above display shows the score assigned to each option for the terminal MAU factor FIELDS OF FIRE. These scores were assigned by the functions f_of_f1 through f_of_f4 shown in Figure 3-2.

Summary of Scores : Textual Display

```
Given 100% of TERRAIN FACTORS factors are solved, best ECOA is DEFEND.
However, SECONDARY ATTACK is nearly as strong.
-----------------------------------------------------------------------------
For this factor, DEFEND has strong support
with a score of 63.3.
The subfactors that provide strong relative support for DEFEND
over other ECOAs are FIELDS OF FIRE,
                     MOBILITY and
                     OBSTACLES.

The subfactors that provide support for other ECOAs over DEFEND
are COVER AND CONCEALMENT,
    KEY FRIENDLY TERRAIN and
    OBSERVATION OF OPFOR.
<More>
                  TERRAIN FACTORS - Subfactor of OPFOR COA
|-Description-----------------|ECOA Scores------------------------------------|
| Wgt: 0.15   #Subfactors:  6 |  PRIMARY    SECONDARY    DEFEND     WITHDRAW   |
|%Sol:  100    #Ans Ques:   8 |    37          58          63          28      |
| Sec:sect 1  #Unans Ques:  0 |                                                |
|-Score Summmarization Menu----------------------------------------------------|
|  <Esc> Quit Menu     1. Textual Summary          4. Cross Sector Summary     |
|      <?> HELP        2. Tabular Summary                                      |
|<PrtSc> Print Screen 3. Numerical Summary                                     |
|------------------------------------------------------------------------------|
```

FIGURE 3-1Ø

The above is a text description describing the results of AI/ENCOA's evaluation of the four options from the perspective of TERRAIN FACTORS. In this display the factors that provide strong relative support for DEFEND are shown. Note that FIELDS OF FIRE is one of these factors.

Summary of Scores : Textual Display

```
Given 100% of TERRAIN FACTORS factors are solved, best ECOA is DEFEND.
However, SECONDARY ATTACK is nearly as strong.
-----------------------------------------------------------------------------
For this factor, SECONDARY ATTACK has moderate support
with a score of 58.3.
The subfactors that provide strong relative support for SECONDARY ATTACK
over other ECOAs are FIELDS OF FIRE,
                    COVER AND CONCEALMENT and
                    KEY FRIENDLY TERRAIN.

The subfactors that provide support for other ECOAs over SECONDARY ATTACK
are MOBILITY,
    OBSERVATION OF OPFOR and
    OBSTACLES.
<More>
                  TERRAIN FACTORS - Subfactor of OPFOR COA
|-Description-----------------|ECOA Scores-----------------------------------|
| Wgt: 0.15   #Subfactors:  6 |  PRIMARY   SECONDARY   DEFEND   WITHDRAW     |
|%Sol:  100   #Ans Ques:   8 |    37         58         65        28        |
| Sec:sect 1  #Unans Ques:  0 |                                              |
|-Score Summmarization Menu--------------------------------------------------|
|  <Esc> Quit Menu    1. Textual Summary          4. Cross Sector Summary    |
|    <?> HELP         2. Tabular Summary                                     |
|<PrtSc> Print Screen 3. Numerical Summary                                   |
|----------------------------------------------------------------------------|
```

FIGURE 3-11

The above is a text description of why SECONDARY ATTACK also has moderate support.

each option for the current MAU factors (the "ECOA Scores" window) and (3) show the menu options.

after "function") name the respective subfactors of the Top-level Factor. The weights (i.e., the relative weights) of all subfactors of a given factor must sum to one. And the "parent" of each subfactor is just the (name of the) factor to which it is a subfactor.

Groups corresponding to the remaining factors are listed in order according to the following rule: the next groups added to the list must correspond to the subfactors of the first factor on the list whose subfactors have not yet had their groups listed.

The format of the groups remains the same, except for Bottom-level Factors. For Bottom-level Factors four additional lines are required. The lines begin "primary_attack ", "secondary_attack ", "defend ", and "withdraw ", respectively. Each of the four lines ends with the identifier, listed before the equal sign in the second line of one of the "function" subgroups, that corresponds to the option named in the beginning of the line. Once again, AI/ENCOA is sufficiently general to permit much more elaborate constructions to be entered; but the present description, which reflects the AI/ENCOA capabilities currently being used, must suffice at the present time.

EXAMPLES OF ALTERING THE RULE BASE

As a simple example of altering the rule base, suppose it is desired to permit Fields of Fire to be characterized in terms of four categories -- 'Greater than 3000 Meters; 'Between 2000 and 3000 Meters', 'Between 1000 and 2000 Meters'; and 'Less than 1000 Meters' -- instead of the present three. Suppose, moreover, that revised scores are to be assigned as follows below.

and Strength of Friendly Division' multiple-choice question; somewhat less complicated forms appear in the groups that follow. The interpretation of these expressions will be apparent to the Pascal programmer; but to try to formulate exact rules for making up and interpreting such expressions would unduly complicate the present document.

Groups and subgroups corresponding to numeric and boolean questions are formed and interpreted similarly. Again note the quasi-Pascal nature of the expressions that are used.

Following the groups corresponding to the questions, and following the names, discussed above, of the options, comes a list of groups beginning with the word "factor" standing alone on a line and ending with a semicolon. This list of "factor" groups continues up until the "end" line. Let us examine the format of each "factor" group, and the order in which the "factor" groups appear, in more detail.

The first "factor" group to be listed corresponds to the Top-level Factor and consists of three additional lines. The first line following "factor" contains some unique, but otherwise arbitrary, Pascal identifier. The second contains the relative weight of the Top-level Factors, which must be one. The third contains the word "parent", followed by a space and then the word "top": this line is obligatory, as both the words "parent" and "top" are code-words recognized by AI/ENCOA and expected here in just the form specified.

Following the group corresponding to the Top-level Factor come the groups corresponding to its subfactors. The format is the same, with the following exceptions. The identifiers (in the first line

corresponds to primary attack, the second such subgroup corresponds to secondary attack, etc.

Within each subgroup is found, immediately after the "function" line (and perhaps some leading spaces), another unique Pascal identifier followed immediately by an equal sign. Identifer and equal sign stand alone on a line. Next come several lines -- as many lines as there were choices with which to answer the multiple-choice question -- of the form: "if [identifier]=[number] then [score] else". Here [identifier] is the same as occured on the line following "multiple-choice question", [number] is one of the numbers preceding the colon which in turn precedes one of the multiple-choice options above the word "function", and [score] is the number of points to be added to the score for the option corresponding to this subgroup in the event that [number] agrees with the number of the choice chosen to answer the multiple-choice question. The final line, "Ø;", indicates the score to be given if the question remains unanswered or is skipped.

Several qualifications need to be added to the preceding paragraph. First, [identifier] could be some other expression than the Pascal identifier given in the second line of the group for this multiple-choice question. AI/ENCOA has the capability of dealing with more complicated expressions in this position; but that capability, though present, is not being exercised at present, and further discussion of it would take us too far afield. Secondly, the order of evaluation of the "if...then...else..." expressions follows the syntax of Pascal. And thirdly, a more complicated form that is in use at the present time occurs in the "function" subgroups within the 'Condition

The group corresponding to the multiple-choice questions starts with the line "multiple_choice_question". The next line contains some legal Pascal identifier; there must be a different identifier for every question. (Note the indentations in the listing: the indentations increase readability, but are not required by the format.) Next comes a line beginning (after, perhaps, some leading spaces) with a hyphen followed immediately by a right caret and then a space, and finally by a phrase in single quotes. AI/ENCOA uses this phrase in making up the question with which it prompts the user. For instance, the first such phrase occurring in the COA rule base is "Fields of Fire"; the question AI/ENCOA asks begins, "Characterize Fields of Fire as". The completion of the question is taken from the following consecutively-numbered lines: note the colon following each number, the use of single quotes to enclose the phrase following the number (and the colon), and the terminal semicolon.

Next within each multiple-choice question group come several subgroups, each starting with the word "function" alone on a line, and each terminated by a semicolon. There is one such subgroup for each option: in the present case, one subgroup for each of four options.

The options themselves are named immediately after the last question group in the list of groups. There the word "score" appears on a line by itself, followed by "options =" on a line by itself, followed by "primary attack,", "secondary attack,", "defend,", and "withdraw;", each on a line by itself. This tells us that we are dealing with four options, named "primary attack", etc., and that the first "function" subgroup in the multiple-choice question group

Intelligent Aid for Estimating Enemy Courses of Action", for more on this subject.)

THE AI/ENCOA RULE BASE

The rule base used for the AI/ENCOA COA demonstration is contained in the file CØA1214.MDL, listed in the "Document of AI/ENCOA Knowledge Base and Source Listing". It is recommended that the reader refer to that document while reading the present subsection. The present appendix discusses the format of the rule base and gives an example of modifying the rule base. Through this description and the accompanying example it is intended to illustrate the power and flexibility of the AI/ENCOA design. Using the methodology illustrated here, an analyst with a suitable technical background should be able to make similar changes to the rule base as the need arises.

The rule base starts with the word "begin" and ends with the word "end", each word standing alone on a line. Line spaces are used frequently throughout the rule base to improve readability; they are ignored by the program processing the rule base.

After the word "begin" there is a long (about twenty pages' worth) sequence of groups of lines with each group corresponding to one question, and with the groups themselves ordered in the same order as that in which the questions would be asked by AI/ENCOA if the user elected to answer all of them.

The format within each group depends on the type of question asked. There are three types of questions that may be asked: multiple-choice questions, numeric questions, and boolean questions. We consider in turn the format of the groups corresponding to each type of question.

of which it is a subfactor, or a sub-subfactor, or a sub-sub-subfactor, etc.

For each factor and for each option, the score for the option is

(1) $\sum_i w_i S_i$

where i ranges over the Bottom-level Factors that are subfactors of the given factor, w_i is the overall weight of the ith Bottom-level Factor, and S_i is the score for the ith Bottom-level Factor for the given option.

The scores and the relative weights are contained within the AI/ENCOA rule base. This rule base also defines the hierarchy of factors. The AI/ENCOA rule base is embodied in a file that is easily accessible to the technical analyst. By suitably altering the rule base, he may readily adapt AI/ENCOA to a wide variety of problems that may appear superficially to have little in common with the particular application discussed in the body of the present document. It is the purpose of this appendix to discuss the structure of the AI/ENCOA rule base and to give several simple ilustrations of how the knowledgeable user may alter the rule base to adapt it to his particular needs.

(We note in passing that the AI/ENCOA rule base is actually much more powerful and flexible than is possible to document fully here. For instance, Equation (1) gives a particularly simple way of combining relative weights and Bottom-level-factor scores to evaluate alternative options. Many other combination rules, well known in the literature of Artificial Intelligence and Decision Analysis, are available to AI/ENCOA, simply through altering the rule base in suitable ways. See the companion Technical Report, "Combining Decision Analysis and Artificial Intelligence Techniques: An

of the proposed field of application.

All this is by way of saying that AI/ENCOA is "generic" software: not limited to just one or two specific applications. In fact, AI/ENCOA has this "generic" versatility in other ways as well. Some of this versatility will become apparent to the reader who examines the source code listing contained in the Document of AI/ENCOA Knowledge Base and Source Listing. Unfortunately, time does not permit a detailed explanation of all the source code that might be of interest to some readers.

OVERVIEW OF AI/ENCOA SCORING

Each factor is assigned a set of scores, one score for each option under consideration. In the present applications, the options happen to be possible enemy courses of action: primary attack, secondary attack, defense, or withdrawal in one application; different avenues of approach in the other application. But AI/ENCOA doesn't really "know" what the options are; it simply "knows" that there are specified options, and that for each factor the score for each option is such and such, based on the answers to the certain questions.

Each factor is also assigned a relative weight. These relative weights are non-negative numbers satisfying the condition: the sum of the relative weights of all factors that are subfactors of the same factor is one. The relative weight of the single Top-level Factor in the hierarchy is defined to be one.

Bottom-level factors are factors which have no subfactors. The "overall weight" of a Bottom-level Factor is defined to be the product of its own relative weight and the relative weights of all the factors

The Generic Nature of AI/ENCOA

INTRODUCTION

AI/ENCOA is in a sense ignorant of the real subject matter with which it deals. What it "sees" is hierarchically-structured factors, with each factor at the same level in the hierarchy being assigned a relative weight; and various scores associated with the answers that the intelligence analyst provides to questions associated with the lowest-level factors (Bottom-level Factors) in the hierarchy. The factors and their interrelationships, and the relative weights, questions, possible answers, and associated scores can all be changed by modifying parameters input to AI/ENCOA, without any need to modify AI/ENCOA itself.

The reason AI/ENCOA works successfully is because of the care and expertise with which the factors were chosen and their interrelationships defined, and the careful consideration given to the choice of scores and relative weights. An expert in Army tactical intelligence analysis and Soviet Doctrine participated extensively in making these decisions. See Appendix B, "Rationale for Score Assignment", of the Artificial Intelligence/Enemy Courses of Action (AI/ENCOA) User's Manual. Moreover, the underlying structure of AI/ENCOA seems well adapted to the class of proble to which the COA and AOA problems belong. By working the same way with experts in other fields, AI/ENCOA could be made to perform similarly for these fields -- provided, of course, that a basic compatibility exists between the "built-in" characteristics of AI/ENCOA and the structure

APPENDIX A

The Generic Nature of AI/ENCOA

for RADC under Contract No.: F30602-83-C-0154, Data Items: A003 and A005, July 13, 1984.

P. Luster, J.R. McIntyre, L. Adelman, P.E. Lehner, M.L. Donnell, "Artificial Intelligence/Enemy Courses of Action (AI/ENCOA) User's Manual," PAR Rpt. No.: 85-07, prepared for Army Research Institute under Contract No.: MDA903-83-C-0311, Data Item 0002AK, February 1985.

E.H. Shortliffe, Computer-Based Medical Consultations: MYCIN. New York: American Alsevier, 1978.

R. Steeb, and S.C. Johnson, "A computer-based interactive system for group decision making," IEEE Transactions on Systems, Man, and Cybernetics, 1981, Vol. SMC-8, No. 8, pp. 544-552.

REFERENCES

L. Adelman, M.L. Donnell, A "Time Lines" Aid and a "Matrices" Aid: Technical Approach Used in Selection Process and General Description of Both Cognitive Support Procedures for Army Tactical Intelligence Analysis. (Status Report) McLean, VA: PAR Technology Corporation, 1983.

L. Adelman, M.L. Donnell, and R.H. Phelps, Intelligence Preparation of the Battlefield: Critique and Recommendations. (Technical Report PR81-4-304) McLean, VA: Decisions and Designs, Inc., 1981.

B.G. Buchanan, et al. "Constructing an Expert System," in Building Expert Systems, eds., Hayes-Roth, F., Waterman, D.A., and Lenat, D.B. Addison-Wesley: Reading, Mass., 1983.

B.G. Buchanan and E.A. Fiegenbaum. Dendral and Meta-Dendral: their application dimension. Artificial Intelligence #11 (1978): 5-24.

R. Davis, "Expert Systems: Where are we? And where do we go from here?", The AI Magazine, Spring, 1982, pp. 3-22.

M.L. Donnell, (SAIC); P.E. Lehner (PAR); F.Rook (PAR); C. Martinez, (BETAC); M. Brenner (Perceptronics); J. Atkinson (KSC); and A. Lazzara (KSC), "Senior Battle Staff Decision Aids Task 1: Planning," Interim Report prepared for RADC under Contract No.: F30602-83-C-0154, March 1983

R.O. Duda, P.E. Hart, and J. Gashnig, "Model Design in the PROSPECTOR Consultant System for Mineral Exploration", in Expert Systems in the Micro-electronic Age (ed. D. Michie) Edinburgh: Edinburgh University Press, 1977, pp. 153-167.

K.R. Hammond, R.L. Cook, and L. Adelman, 1977. "POLICY: An aid for decision making and international communication," Columbia Journal of World Business, Vol. 12, No. 3, pp. 79-93.

K.R. Hammond, G.H. McClelland, and J. Mumpower, Human Judgment and Decision-Making: Theories, Methods, and Procedures. New York, NY: Hemisphere/Praeger, 1980.

R.L. Keeney, and H. Raffai, Decisions with Multiple Objectives: Preference and Value Tradeoffs. New York: John Wiley & Sons, Inc., 1976.

P.E. Lehner, L. Adelman, Matthew A. Probus, Jackson O. Crowley, Michael L. Donnell, Joel H. Krenis, "Enemy Courses of Action Evaluation Aid: Functional Description and Design Plan Senior Battle Staff Decision Aids," PAR Rpt. 84-70, prepared

to allow questions to be answered from sources other than the user (e.g., direct data base access). In the ENCOA models, for instance, most of the questions request information that should be resident in an order of battle (OB) data base or situation map. Consequently, with this latter enhancement the analyst would not be burdened with the need to answer most of the AI/ENCOA questions.

The third direction is based on combining the AI/ENCOA work with that of two other efforts. In related efforts, (Lehner, et al., 1984; Donnell, et al., 1983) the COA problem is analyzed at the CORPS level to identify, across an entire front, which CORPS sectors the enemy commander will perceive as the most critical or important to reinforce and resupply. Second, a version of a timelines aid (Adelman and Donnell, 1983) allows estimates of the amount of reinforcement and resupply that could be delivered to different sectors over different timeframes. Combining these two aids and AI/ENCOA into a single system would provide for an integrated COA evaluation system that would allow an analyst to examine the ENCOA problem from a number of different perspectives. For example, combining AI/ENCOA with the timelines aid would provide an analyst with an ability to project future enemy COAs and AOAs given different scenarios based on the sectors that will receive heaviest support.

sufficient knowledge engineering with multiple experts and scenario testing, AI/ENCOA could be enhanced to adaptively adjust to alternative problem contexts (alternative enemy types, different terrain conditions, alternative political contexts, etc.)

The second direction involves enhancements to the generic AI/MAU software. Such enhancements would focus primarily on improving output/result displays and expanding the syntax of the rule base. Regarding output displays, AI/ENCOA is like most systems in that it relies on a set of standard display formats. The content and format of the displays remain static and are not adaptive to either user characteristics or the specifics of the present problem. A <u>rule-based</u> <u>display controller</u> could be added to the system to provide for adaptive displays. The display controller would operate in a manner similar to the parameter assignment rules, where the preconditions would correspond to questions and the post conditions would correspond to display control actions. For example, if FIELDS OF FIRE and OBSTACLES were the primary factors discriminating Primary Attack from Defend, then a display control action might be to show a terrain map with these factors highlighted. In this way, the rules for display control would reside and interact with the domain specific composition rules.

Improvements to the rule base syntax would include expanding the number of different question types (e.g., Estimate the likelihood that...?") and function types (e.g., matrix multiplication) that could be entered into the rule base. This capability could be used to reduce the four-function mapping shown in Figure 3-2 into a single matrix mapping. Related to this would be an expansion of the syntax

4.0 SUMMARY AND DIRECTIONS FOR FUTURE WORK

AI/ENCOA reflects several advances over the previous ENCOA system that use only multi-attribute utility theory. First, the combined AI/MAU software allows a user interface that queries tactical intelligence analysts in terms and references familiar to them. AI/ENCOA will, for instance, query the user about the number of enemy divisions within 15km of the front. ENCOA, on the other hand, would ask users to rate enemy division strength on a 0 to 100 scale. Second, the AI/ENCOA software is entirely generic, allowing users to develop or tailor models for any type of option selection domain. ENCOA, on the other hand, was specific to the COA problem. Finally, the original ENCOA only addressed the problem of evaluating alternative courses of action (COAs). AI/ENCOA, on the other hand, contains two models. The first model evaluates, at the division level, whether the enemy commander might engage in a Primary Attack, Secondary Attack, Defense or Withdrawal. The second model discriminates between primary and secondary AOAs given that some form of attack will occur.

There are three future directions that AI/ENCOA related activities could take. These are individually discussed below.

The first direction involves significant additional knowledge engineering. The models presently in AI/ENCOA have fixed weights that are consistent with the perspective likely to be taken by a Soviet commander fighting on European terrain. The AI/MAU software, however, allows dynamic weight assignment and model restructuring on the basis of the characteristics of the decision problem. Consequently, with

'Fields of Fire' Range	Scores Primary Attack	Secondary Attack	Defend	Withdraw
Greater than 3000 Meters	10	40	100	0
Between 2000 and 3000 Meters	80	85	90	45
Between 1000 and 2000 Meters	90	95	35	55
Less than 1000 Meters	100	100	25	65

Then the revised listing contains the "group of lines" shown in Exhibit A-1.

Comparing Exhibit A-1 with the original listing, we see changes in the limits specified for choices 2 and 3, and the addition of a fourth choice, "4: 'Less than 1000 Meters'". Moreover, each "function" subgroup contains an additional line beginning, "if f_of_f=4 then". The method for inserting the revised scores is obvious upon comparing Exhibit A-1 with the text.

As a second example of altering the rule base, suppose it is desired to add the factor KEY ENEMY TERRAIN as a subfactor of TERRAIN FACTORS. To do so will require specifying a relative weight for KEY ENEMY TERRAIN and readjusting the relative weights for the other subfactors of TERRAIN FACTORS so that the sum of the relative weights of all these subfactors remains one. For simplicity, suppose that the relative weight for KEY ENEMY TERRAIN is to be 0.14, the revised relative weight for key friendly terrain is to be 0.15, and all other relative weights are to remain the same.

Two insertions must be made in the rule base, and one alteration of information originally in the rule base. The first change is the insertion, immediately after the "group of lines" corresponding to the multiple-choice question "Characterize Key

```
multiple_choice question
    f_of_f
    -> 'Fields of Fire'
    1: 'Greater than 3000 Meters'
    2: 'Between 2000 and 3000 Meters'
    3: 'Between 1000 and 2000 Meters'
    4: 'Less than 1000 Meters';

function
    f_of_f1=
    if f_of_f=1 then 10 else
    if f_of_f=2 then 80 else
    if f_of_f=3 then 90 else
    if f_of_f=4 then 100 else
    0;

function
    f_of_f2=
    if f_of_f=1 then 40 else
    if f_of_f=2 then 85 else
    if f_of_f=3 then 95 else
    if f_of_f=4 then 100 else
    0;

function
    f_of_f3=
    if f_of_f=1 then 100 else
    if f_of_f=2 then 90 else
    if f_of_f=3 then 35 else
    if f_of_f=4 then 25 else
    0;

function
    f_of_f4=
    if f_of_f=1 then 0 else
    if f_of_f=2 then 45 else
    if f_of_f=3 then 55 else
    if f_of_f=4 then 65 else
    0;
```

EXHIBIT A-1

ILLUSTRATING A REVISION OF A PORTION OF THE RULE BASE

Friendly Terrain as ...", of the "group" of lines shown in Exhibit A-2. This "group" of lines will allow the user to choose "1: 'Critical to Enemy Operations'", etc., as answers to the question. The resulting scores will be determined as follows:

'Key Enemy Terrain'	Scores: Primary Attack	Secondary Attack	Defend	Attack
Critical to Enemy Operations	90	50	10	0
Advantageous but not Critical to Enemy Operations	50	70	30	0
No Key Terrain	30	70	50	0

The second insertion and the alteration are shown in Exhibit A-3. The change of relative weight for KEY FRIENDLY TERRAIN from 0.29 to 0.15 is shown on the third line of this exhibit. This is the only change made to the "factor" group corresponding to KEY FRIENDLY TERRAIN.

The inserted "factor" group of lines corresponding to KEY ENEMY TERRAIN is also shown in Exhibit A-3. The meaning of the lines "weight 0.14" and parent terrain factors" is clear. Finally, note that mention of the four functions ket1 through ket4 in the final four lines of the "group" is the mechanism whereby the scores shown in Exhibit A-2 (weighted by the appropriate weighting factor) are added to the appropriate ECOA Score while AI/ENCOA is being run.

THE MECHANICS OF ALTERING THE RULE BASE

Any standard editor -- EDLIN, for example -- may be used to modify the "Courses of Action" rule base, COA1124.MDL, located in the directory AIENCOA.

```
multiple_choice question
    ket
    -> 'Key Enemy Terrain'
    1: 'Critical to Enemy Operations'
    2: 'Advantageous but not Critical to Enemy Operations'
    3: 'No Key Terrain';

function
    ket1=
    if ket=1 then 90 else
    if ket=2 then 50 else
    if ket=3 then 30 else
    0;

function
    ket2=
    if ket=1 then 50 else
    if ket=2 then 70 else
    if ket=3 then 70 else
    0;

function
    ket3=
    if ket=1 then 10 else
    if ket=2 then 30 else
    if ket=3 then 50 else
    0;

function
    ket4=
    0;
```

EXHIBIT A-2

THE "GROUP" OF LINES CORRESPONDING TO THE INSERTED MULTIPLE-CHOICE QUESTION REGARDING KEY ENEMY TERRAIN

```
factor
    key_friendly_terrain
    weight 0.15
    parent terrain_factors
    primary_attack kft1
    secondary_attack kft2
    defend kft3
    withdraw kft4;

factor
    key_enemy_terrain
    weight 0.14
    parent terrain_factors
    primary_attack ket1
    secondary_attack ket2
    defend ket3
    withdraw ket4;
```

EXHIBIT A-3

CHANGES AND INSERTIONS IN THE RULE-BASE "FACTOR" GROUPS TO ACCOMMODATE KEY ENEMY TERRAIN

To convert the modified file into the internal forms needed by AI/ENCOA, it must be processed by the rule-base compiler. Working within the AIENCOA directory, the rule-base compiler is started by typing **compile**. The compiler will clear the screen, print a message, and prompt for a file name. Enter the name of the model file without the "MDL" extension: i.e., enter **COA1124,** in the present case. The compiler will produce files with the name you entered and the extensions ".FCT", ".FUN", ".QUE", ".ANS", ".TRM", and ".OTH". These files are not in human-readable form, but are used automatically by the AI/ENCOA Program.

The "Avenues of Approach" rule base, AOA1116.MDL. may be modified similarly.